

## Claims

- [c1] What is claimed is:
1. A method of managing an input buffer in a media player for playing a media file, the media file comprising a stream of frames, each frame having at least a main\_data field containing encoded media samples and a main\_data\_begin field indicating an overflow of the main\_data field, the media player including a parser, an input buffer, a decoder, and a totalizer, the parser is capable of parsing the stream of frames to the decoder and informing the decoder whether to decode from the beginning of the media file, or from the middle of the media file, the method comprising:  
if the decoder is informed to decode from the middle of the media file, then:  
locating a first frame having a first main\_data\_begin field and a first main\_data field, if a value in the totalizer is less than a value in the first main\_data\_begin field, adding a size of the first main\_data field to the totalizer, and storing the first main\_data field in the input buffer;  
locating a second frame which is downstream to the first frame, the second frame having a second main\_data\_begin field and a second main\_data field, if a value in the totalizer is equal to or larger than a value in the second main\_data\_begin field, decoding the stream of frames starting from the second frame using both the first main\_data field stored in the input buffer and the second main\_data field;  
if the decoder is informed to decode from the beginning of the media file, then  
locating a third frame having a third main\_data\_begin field with a value of zero and a third main\_data field, and decoding the stream of frames starting from the third frame.
  - [c2] 2. The method of claim 1 wherein the media file is an MP3 file.
  - [c3] 3. The method of claim 1 wherein the totalizer is initialized to zero.
  - [c4] 4. A method of managing an input buffer in a media player for playing a media file, the media file comprising a stream of frames, each frame having at least a main\_data field containing encoded media samples and a main\_data\_begin field indicating an overflow of the main\_data field, the media player including a

[illegible]

locating a first frame having a first main\_data\_begin field and a first main\_data field, if a value in the totalizer is less than a value in the first main\_data\_begin field, adding a size of the first main\_data field to the totalizer, and storing the first main\_data field in the input buffer;

locating a second frame which is downstream to the first frame, the second frame having a second main\_data\_begin field and a second main\_data field, if a value in the totalizer is equal to or larger than a value in a second main\_data\_begin field, decoding the stream of frames starting from the second frame using both the first main\_data field stored in the input buffer and the second main\_data field.

- [c5] 5. The method of claim 4 wherein the media file is an MP3 file.
- [c6] 6. The method of claim 4 wherein the totalizer is initialized to zero. ◆
- [c7] 7. A method of managing an input buffer of a playback control for playing an MP3 (Motion Pictures Experts Group Layer III Audio) file on an MP3 player, the MP3 file comprising a sequential series of frames containing data, the method comprising:  
locating a downstream frame, if a value in a totalizer is less than a value in a main\_data\_begin field of the frame, adding a calculated size of a main\_data of the frame to the totalizer, and storing the main\_data of the frame in the input buffer for later referencing.
- [c8] 8. The method of claim 7 further comprising reading an error check field if the error check field is present in the frame, and using the error check field to verify integrity of data within the frame.
- [c9] 9. The method of claim 7 wherein the input buffer comprises a memory accessible by the playback control.
- [c10] 10. The method of claim 7 wherein the playback control selects a parser or the playback control selects the input buffer as a source of audio data to be processed and played.

- [c11] 11. The method of claim 7 wherein the totalizer is initialized to zero.
- [c12] 12. The method of claim 7 further comprising using a variable to indicate that a starting frame has been located.
- [c13] 13. The method of claim 12 wherein the variable is of a Boolean type.
- [c14] 14. The method of claim 7 further comprising decoding a header of the frame.
- [c15] 15. The method of claim 7 further comprising decoding an audio data of the frame.
- [c16] 16. The method of claim 7 further comprising locating a synchronization word of the frame.
- [c17] 17. An MP3 (Motion Pictures Experts Group Layer III Audio) player for decoding and outputting MP3 files, the MP3 player comprising;  
an input buffer for storing main\_data of an MP3 frame for later referencing;  
a playback control for managing the input buffer, the playback control being capable of determining whether all of the main\_data required by the MP3 frame has been stored in the input buffer.
- [c18] 18. The MP3 player of claim 17 further comprising a totalizer used by the playback control for determining size of the main\_data currently in the input buffer and in the MP3 frame.
- [c19] 19. The MP3 player of claim 17 wherein decoding the main\_data of the frame is processed using the Huffman decoding method.